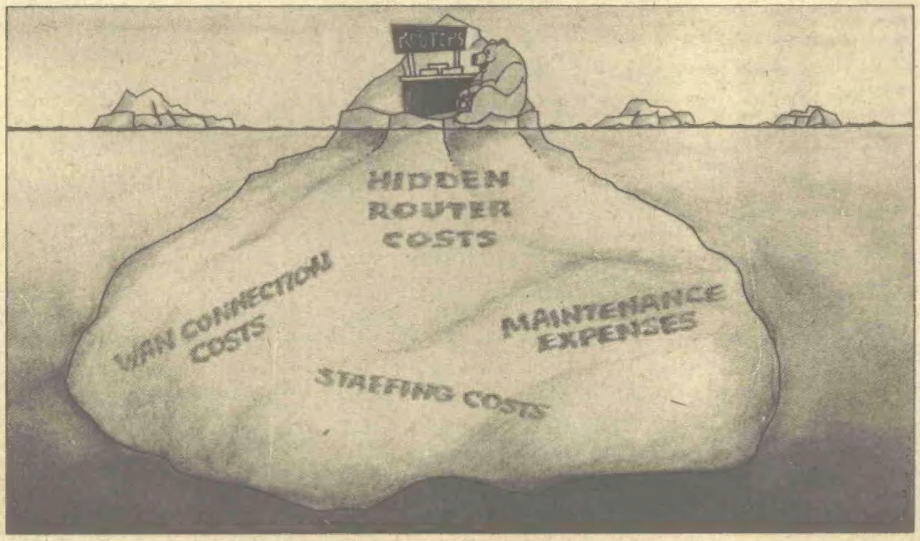
Clipper

เมษ พงษ์อำไพ

คลิปเปอร์ TIP



เสนอฟังก์ชันต่าง ๆ ที่ใช้ในการพัฒนาโปรแกรมสำหรับคลิปเปอร์ที่เป็นภาษาไทย

คลิปเปอร์เป็นภาษาฐานข้อมูลที่มีผู้นิยมใช้ในประเทศไทยเป็นจำนวนมาก (สังเกตได้จากตามร้านหนังสือคอมพิวเตอร์มีหนังสือเกี่ยวกับคลิปเปอร์ทั้งภาษาไทยและภาษาอังกฤษมากมาย) โดยเฉพาะโปรแกรมเมอร์อาชีพ สาเหตุอาจเนื่องมาจากคลิปเปอร์เป็นคอมพายเลอร์ซึ่งเหมาะกับการพัฒนาโปรแกรมประยุกต์เฉพาะงาน เพื่อส่งมอบต่อลูกค้าโดยไม่ต้องให้ซอร์สโค้ด โปรแกรมที่เขียนและคอมพายล์แล้วสามารถนำไปรันในเครื่องใด ๆ ก็ได้ โดยไม่จำเป็นต้องมีคลิปเปอร์อยู่ในเครื่อง (เป็น STAND ALONE EXE) ทำให้เสียเงินซื้อคลิปเปอร์ชุดเดียว สามารถพัฒนาโปรแกรมเพื่อรันในเครื่องหลาย ๆ เครื่องได้โดยไม่เสียค่าลิขสิทธิ์เพิ่ม ซึ่งขณะนี้มีข่าวว่ารัฐบาลกำลังผลักดันกฎหมายลิขสิทธิ์ผ่านสภา โปรแกรมเมอร์ภาษา xBASE ทั้งหลายควรเตรียมตัวไว้แต่เนิ่น ๆ เพื่อรอรับสถานการณ์และการเปลี่ยนแปลงที่เกิดจากกฎหมายฉบับนี้

คลิปเปอร์เป็นภาษาที่มีความยืดหยุ่นสูงมาก มีคำสั่งและฟังก์ชันให้เลือกใช้มากมาย นอกจากนั้นยังสามารถสร้างโพรซีเยอร์และฟังก์ชันขึ้นเองด้วยภาษาคลิปเปอร์ หรือซี หรือแอสเซมบลีก็ได้ โพรซีเยอร์หรือฟังก์ชันที่สร้างไว้สามารถนำไปรวมกับไลบรารีของคลิปเปอร์เองเพื่อเก็บไว้ใช้เป็นฟังก์ชันมาตรฐานสำหรับพัฒนาโปรแกรมอื่น ๆ ต่อไป

ในฉบับนี้ผู้เขียนขอเสนอฟังก์ชันบางส่วนที่ได้เขียนเอาไว้สำหรับใช้ในการพัฒนาโปรแกรมภาษาไทย เนื่องจากฟังก์ชันที่มีอยู่เดิมในคลิปเปอร์ไม่เหมาะสมหรือไม่มีฟังก์ชันที่ต้องการ ฟังก์ชันทั้งหมดที่นำเสนอเขียนด้วย CA CLIPPER 5.2

TLEN( ) นับจำนวนคอลัมน์ของอักษรไทย

เนื่องจากฟังก์ชัน LEN( ) จะนับจำนวนตัวอักษรที่มีอยู่ทั้งหมดในสตริง ซึ่งถ้าเป็นภาษาอังกฤษล้วนก็จะเท่ากับจำนวนคอลัมน์ที่ใช้ แต่ถ้าเป็นภาษาไทยจะพบว่ามีปัญหาที่สระ วรรณยุกต์ ที่อยู่ข้างบนหรือข้างล่างบรรทัด ทำให้นับจำนวนคอลัมน์ได้มากกว่าความเป็นจริง

ฟังก์ชัน TLEN( ) ดังรูปที่ 1 จะนับจำนวนคอลัมน์ภาษาไทยได้อย่างถูกต้อง โดยไม่นับอักษรที่อยู่บนหรือล่างบรรทัด การนำไปใช้ท่านต้องกำหนดว่ารหัสแอสกีใดที่จะไม่ให้นับ โดยใส่คำสั่ง

```
#DEFINE THAI_UPDN
```

ไว้ที่ต้นโปรแกรมหลัก ขึ้นอยู่กับว่าท่านใช้รหัสเกษตร หรือ สมอ. หรือรหัสภาษาไทยอื่น ๆ ตามตัวอย่างในรูป 1 ใช้รหัสเกษตร แต่ได้แสดงตัวอย่างรหัส สมอ.แบบหนึ่งไว้แล้ว (รหัส สมอ. ก็มีหลายแบบ)

```
#DEFINE THAI_UPDN CHR(224)+CHR(225)+CHR(226)+CHR(227)+CHR(228)+;
                  CHR(215)+CHR(216)+CHR(217)+CHR(218)+CHR(219)+;
                  CHR(220)+CHR(221)+CHR(222)+CHR(223)
*    CODE SMO     CHR(209)+CHR(212)+CHR(213)+CHR(214)+CHR(215)+;
*                 CHR(216)+CHR(217)+CHR(218)+CHR(231)+CHR(232)+;
*                 CHR(233)+CHR(234)+CHR(235)+CHR(236)+CHR(237)
?"สู่สุขาวดี"
?" LEN()=",  LEN("สู่สุขาวดี")
?"TLEN()=", TLEN("สู่สุขาวดี")
RETURN

FUNC TLEN(THAI_STR)
LOCAL CHR_NO, ALL_LEN:=0
FOR CHR_NO:=1 TO LEN(THAI_STR)
  IF(SUBS(THAI_STR,CHR_NO,1)$THAI_UPDN,"",ALL_LEN++)
NEXT
RETURN ALL_LEN
```

รูปที่ 1 TLEN( ) นับจำนวนคอลัมน์ภาษาไทย

TPADR( ), TPADC( ), TPADL( ) ยืดความยาวสตริง

CLIPPER 5.2 มีฟังก์ชัน PADR( ), PADC( ), PADL( ) สำหรับเพิ่มช่องว่าง (หรืออักษรใด ๆ) ที่ข้างขวาหรือข้างซ้ายของสตริงเพื่อให้สตริงนั้นมีความยาว (จำนวนคอลัมน์) ตามที่กำหนด เหมาะสำหรับจัดรูปแบบที่แสดงบนจอภาพหรือพิมพ์ออกเครื่องพิมพ์

รูปแบบการใช้ฟังก์ชัน TPADR( ), TPADC( ), TPADL( ) เหมือนกับ PADR( ), PADC( ), PADL( ) ทุกประการ ดังตัวอย่างในรูปที่ 2 พารามิเตอร์มี 3 ตัวดังนี้

TPADR (นิพจน์ตัวอักษร ความยาวที่ต้องการ ตัวอักษรที่จะเพิ่ม)

เช่น TPADR ("ตัวอย่าง", 9, "R") จะได้ค่าเป็น ตัวอย่างRRR

TPADC ("ตัวอย่าง" ,12, "X") จะได้ค่าเป็น XXXตัวอย่างXXX

ท่านสามารถส่งพารามิเตอร์ 2 หรือ 3 ตัวก็ได้ ถ้าไม่ส่งตัวที่สามถือว่าใช้ช่องว่างแทน

จากโปรแกรมรูปที่ 2 จะเห็นว่า TPADR( ), TPADC( ), TPADL( ) จะเรียกใช้ TPAD( ) และ TPAD( ) จะไปเรียก TLEN( ) อีกที ฉะนั้นจะต้องมี TPAD( ) และ TLEN( ) อยู่ในโปรแกรมด้วย (หรืออาจจะเอาฟังก์ชันทั้งหมดลิงค์ไว้ในไลบรารีมาตรฐานของคลิปเปอร์เลยก็ได้)

ตัวอย่างการนำไปใช้งานมีหลายแบบ เช่น ถ้าเราต้องการพิมพ์ฟิลด์ 2 ฟิลด์คือ NAME และ AGE ด้วยคำสั่ง

```
LIST NAME+STR(AGE, 3) TO PRINT
```

โดย NAME นั้นมีภาษาไทยอยู่ด้วย จะทำให้ AGE เยื้องไปมา ถ้าเซตระบบภาษาไทยให้ชดเชยสระเมื่อพบช่องว่าง 2 ช่อง ก็จะต้องใช้คำสั่ง

```
LIST NAME+" "+STR(AGE, 3) TO PRINT
```

ซึ่งจะทำให้มีช่องว่างระหว่างฟิลด์มากเกินความจำเป็น ยิ่งถ้ามีฟิลด์ที่ต้องพิมพ์หลาย ๆ ฟิลด์ ความกว้างของกระดาษอาจจะพิมพ์ไม่พอ การควบคุมความกว้างทำได้ลำบาก ดังนั้นผู้เขียนจึงใช้วิธีเซตระบบภาษาไทยให้ชดเชยสระเมื่อพบ CHR(160) 1 ตัว เมื่อไรก็ตามที่เราไม่ต้องการให้ชดเชยสระก็ไม่ต้องส่ง CHR(160) ออกเครื่องพิมพ์ ทำให้ไม่ต้องคอยพะวงว่าเครื่องจะชดเชยเองตอนพบช่องว่าง (ทั้ง ๆ ที่เราไม่ต้องการให้ชดเชย) แต่เราต้องควบคุมความกว้างแต่ละฟิลด์เองด้วย TPADR( ) ดังตัวอย่าง

```
LIST TPADR(NAME, 23)+STR(AGE, 3) TO PRINT
```

อีกตัวอย่างหนึ่งในการนำไปใช้คือใช้ในคำสั่ง PROMPT ดังตัวอย่าง

```
#DEFINE THAI_UPDN CHR(224)+CHR(225)+CHR(226)+CHR(227)+CHR(228)+;
                  CHR(215)+CHR(216)+CHR(217)+CHR(218)+CHR(219)+;
                  CHR(220)+CHR(221)+CHR(222)+CHR(223)
CLEAR
SET COLOUR TO "R/W"
@ 3,30 SAY TPADC("บรรทัดนี้วางกลาง", 20)
@ 5,30 SAY TPADL("ชิดขวา", 20)
@ 7,30 SAY TPADR("ชิดซ้าย", 20)
@ 9,30 SAY TPADC("ทุกบรรทัดยาวเท่ากัน",20)
RETURN

FUNC TPADR(THAI_STR, THAI_LEN, FILL_CHR)
RETU TPAD(THAI_STR, THAI_LEN, 1, IF(PCOUNT()<3, " ", FILL_CHR))

FUNC TPADC(THAI_STR, THAI_LEN, FILL_CHR)
RETU TPAD(THAI_STR, THAI_LEN, 2, IF(PCOUNT()<3, " ", FILL_CHR))

FUNC TPADL(THAI_STR, THAI_LEN, FILL_CHR)
RETU TPAD(THAI_STR, THAI_LEN, 3, IF(PCOUNT()<3, " ", FILL_CHR))

FUNC TPAD(THAI_STR, THAI_LEN, TYPE_PAD, FILL_CHR)
LOCAL LEN_PARA:=TLEN(THAI_STR), RETURN_STR, LEN_PRECHR
IF EMPTY(THAI_STR)
   RETU SPACE(THAI_LEN)
ENDI
DO WHILE LEN_PARA > THAI_LEN
   LEN_PARA:=TLEN(THAI_STR:=LEFT(THAI_STR, LEN(THAI_STR);
            - (LEN_PARA-THAI_LEN)))
ENDD
DO CASE
CASE TYPE_PAD=1
   RETURN_STR:=THAI_STR + REPL(FILL_CHR, THAI_LEN-LEN_PARA)
CASE TYPE_PAD=3
   RETURN_STR:=REPL(FILL_CHR, THAI_LEN-LEN_PARA) + THAI_STR
OTHE
   LEN_PRECHR:=INT((THAI_LEN-LEN_PARA)/2)
   RETURN_STR:=REPL(FILL_CHR, LEN_PRECHR) + THAI_STR +;
               REPL(FILL_CHR, THAI_LEN-LEN_PARA-LEN_PRECHR)
ENDC
RETURN RETURN_STR


FUNC TLEN(THAI_STR)
LOCAL CHR_NO, ALL_LEN:=0
FOR CHR_NO:=1 TO LEN(THAI_STR)
  IF(SUBS(THAI_STR,CHR_NO,1) $ THAI_UPDN, "", ALL_LEN++)
NEXT
RETURN ALL_LEN
```

รูปที่ 2 TPADR( ), TPADC( ), TPADL( ) เพิ่มความกว้างตัวอักขระ

```
@ 3, 30 PROMPT TPADC ("1.เพิ่มข้อมูล", 20)
@ 4, 30 PROMPT TPADC ("2.พิมพ์รายงาน", 20)
MENU TO I
```

จะทำให้เมื่อผู้ใช้โปรแกรมเลื่อนแถบแสง แถบแสงจะยังคงมีความยาวเท่าเดิม

TDTOC( ), TCTOD( ) แปลงไปมาระหว่างอักขระกับวันที่

การเก็บข้อมูลชนิดวันที่ในคลิปเปอร์จะใช้รูปแบบ MM/DD/YY (ถือว่า SET DATE AMERICAN) แต่โปรแกรมภาษาไทยส่วนใหญ่ต้องการกำหนดรูปแบบเป็น DD/MM/YY โดยที่ YY คือตัวเลข 2 ตัวหลังของปีพ.ศ.

ดังนั้นผู้เขียนจึงเขียนฟังก์ชันสำหรับแปลงรูปแบบวันที่แบบไทยดังรูปที่ 3 โดยที่ข้อมูลที่เก็บในฟิลด์แบบวันที่ยังคงเก็บแบบเดิมของคลิปเปอร์ แต่เวลาเอามาแสดงบนจอภาพ หรือออกเครื่องพิมพ์ หรือรอรับข้อมูลจากผู้ใช้ จะให้แสดงแบบ วัน/เดือน/พ.ศ. โดยใช้ TDTOC( ) ดังตัวอย่าง

```
? "วันเกิดคือ"+TDTOC(BORN)
```

โดยที่ BORN คือฟิลด์หรือตัวแปรแบบวันที่เก็บวันเกิด (MM/DD/YY)

เมื่อเวลาโปรแกรมหยุดรอรับค่าวันที่เพื่อเก็บในฟิลด์แบบวันที่ก็รอรับตัวอักษรตามรูปแบบไทย DD/MM/YY แล้วใช้ TCTOD( ) แปลงเป็นแบบวันที่เพื่อเก็บในฟิลด์ดังตัวอย่างในรูปที่ 4

ผู้ใช้สามารถใส่วันเกิดแบบไทย เช่น 18/04/03 (18 เม.ย. 2503) โปร-

```
CLEAR
DATE_CHR=SPACE(8)
@ 22,0 SAY "THAI DATE =" GET DATE_CHR PICT "99/99/99"
READ
DATE_VAR:=TCTOD(DATE_CHR)
?"CONVERT TO DATE TYPE =",DATE_VAR," FOR REPLACE DATE-FIELD"
?"CONVERT TO CHAR TYPE =",TDTOC(DATE_VAR)," FOR DISPLAY ON SCREEN"
RETU

FUNC TCTOD(TCHR_DATE)
RETU CTOD(SUBS(TCHR_DATE,4,3)+SUBS(TCHR_DATE,1,3)+;
     RIGHT(STR(VAL(SUBS(TCHR_DATE,7,2))+57),2))

FUNC TDTOC(TDATE_CHR)
RETU IF(EMPTY(TDATE_CHR), "  /  /  ";;
        STRTRAN(STR(DAY(TDATE_CHR),2)," ","0") + "/" +;
        STRTRAN(STR(MONTH(TDATE_CHR),2)," ","0") + "/" +;
        STRTRAN(STR((YEAR(TDATE_CHR)+543)%100,2)," ","0"))
```

รูปที่ 3 แปลงตัวอักษรวันที่แบบไทยเป็น DATE TYPE ด้วย TCTOD( ) แปลง DATE TYPE เป็นตัวอักษรวันที่แบบไทยด้วย TDTOC( )

```
TBORN := TDTOC(BORN)
@ 5, 10 SAY "โปรดใส่วันเกิด (วว/ดด/ปป)" GET TBORN PICT "99/99/99"
READ
REPL BORN WITH TCTOD(TBORN)
```

รูปที่ 4 ตัวอย่างการแปลงตัวอักษรวันที่แบบไทยเป็น DATE TYPE เพื่อเก็บในฟิลด์

```
FOR I=0 TO 24
  @ I,0 SAY STR(I, 2) + " " + REPL(CHR(65+I), 76)
NEXT
SCREENATTB(1, 4, 23, 77, "N/N")
INKEY(0)
SCREENATTB(1, 4, 23, 77, SETCOLOR())
SCREENATTB(10, 20, 15, 60, "G*/N")
SCREENATTB(11, 21, 12, 59, "B/RB")
SCREENATTB(13, 21, 14, 59, "GR/R")
INKEY(0)

FUNC SCREENATTB(ROW1, COL1, ROW2, COL2, COLOURSET)
LOCAL LASTROW:=ROW(), LASTCOL:=COL()
LOCAL OLDSCRN:=SAVESCREEN(ROW1, COL1, ROW2, COL2)
@ ROW1, COL1 SAY " " COLOR (COLOURSET)
SETPOS(LASTROW, LASTCOL)
RESTSCREEN(ROW1, COL1, ROW2, COL2, TRANSFORM(OLDSCRN, ;
          REPLICATE( "X"+SUBS(SAVESCREEN(ROW1, COL1, ;
          ROW1, COL1), 2, 1) , LEN(OLDSCRN)/2 ) ))
RETURN NIL
```

รูปที่ 5 SCREENATTB( ) เปลี่ยนสีตัวอักษรบนจอภาพ

```
FUNC NUM_LETTER(NUM_LETTER)
IF NUM_LETTER = 0 ; RETU "ศูนย์บาท" ; ENDI
PRIVATE NUM_NAME:={ "หนึ่ง","สอง","สาม","สี่","ห้า","หก","เจ็ด","แปด","เก้า" }
PRIVATE TEN_NAME:={ "สิบ","ยี่สิบ","สามสิบ","สี่สิบ","ห้าสิบ","หกสิบ","เจ็ดสิบ","แปดสิบ","เก้าสิบ" }
PRIVATE DIG_NAME:={ "แสน","หมื่น","พัน","ร้อย","สิบ","" }, NUM_STR:=STR(NUM_LETTER,15,2)
PRIVATE LETTER_RET := IF(!EMPTY(LEFT(NUM_STR,6)), NUM_LET2(LEFT(NUM_STR,6)) + "ล้าน", "")
LETTER_RET += NUM_LET2(SUBS(NUM_STR,7,6)) + IF(NUM_LETTER>=1, "บาท", "")
RETU LETTER_RET + IF(SUBS(NUM_STR,14)="00", "ถ้วน", NUM_LET2(SPACE(4)+SUBS(NUM_STR,14)) + "สตางค์")

FUNC NUM_LET2(NUM_LET2)
LOCAL DIGIT, DIGIT_VAL, K:=""
FOR DIGIT:=1 TO 6
  DIGIT_VAL := VAL(SUBS(NUM_LET2, DIGIT, 1))
  K += IF(DIGIT_VAL>0, IF(DIGIT=6.AND.DIGIT_VAL=1.AND.VAL(NUM_LET2)>1, "เอ็ด",;
       IF(DIGIT<>5, NUM_NAME[DIGIT_VAL] + DIG_NAME[DIGIT], TEN_NAME[DIGIT_VAL])), "")
NEXT
RETU K
```

รูปที่ 6 NUM_LETTER แปลงตัวเลขเป็นตัวอักษรคำอ่าน

แกรมจะแปลงเป็น 04/18/60 เพื่อเก็บในฟิลด์ BORN

SCREENATTB( ) ปรับแอตทริบิวต์ของจอภาพ

บางครั้งเราต้องการให้ตัวอักษรที่มีอยู่เดิมบนจอเปลี่ยนสีหรือทำให้ตัวอักษรกระพริบโดยยังคงตัวอักษรแบบเดิมเอาไว้ (เปลี่ยนแต่แอตทริบิวต์) คลิปเปอร์ไม่มีฟังก์ชันแบบนี้ผู้เขียนจึงเขียนขึ้นมาใช้เองดังรูปที่ 5

พารามิเตอร์ที่ส่งผ่านให้ SCREENATTB( ) คือ (แถวบน คอลัมน์ซ้าย แถวล่าง คอลัมน์ขวา รูปแบบของสี) โดยรูปแบบของสีใช้เช่นเดียวกับที่กำหนดด้วย SETCOLOR( )

NUM_LETTER( ) แปลงตัวเลขเป็นตัวอักษรคำอ่านตัวเลขนั้น

เวลาเขียนโปรแกรมสำหรับพิมพ์ใบเสร็จมักต้องมีคำอ่านจำนวนเงินเป็นตัวหนังสือ เช่น 231.25 ก็พิมพ์คำว่าสองร้อยสามสิบเอ็ดบาทยี่สิบห้าสตางค์ในช่องที่กำหนดให้ใส่ตัวหนังสือ

ฟังก์ชันตามรูปที่ 6 สามารถแปลงตัวเลขเป็นคำอ่านภาษาไทย ดังตัวอย่าง

```
?NUM_LETTER(NET)
```

โดย NET เป็นตัวแปรหรือฟิลด์แบบตัวเลข ถ้าต้องการพิมพ์ให้ลงตรงกลางของพื้นที่ที่กำหนดในกระดาษ อาจใช้ TPADC( ) ร่วมด้วย เช่น ถ้าพื้นที่ที่จะพิมพ์กว้าง 4 นิ้ว และพิมพ์ 12 ตัว/นิ้ว อาจใช้คำสั่ง

```
?TPADC(NUM_LETTER(NET), 48)
```

สรุป

จากตัวอย่างฟังก์ชันที่นำมาเสนอนี้จะเห็นว่า CLIPPER 5.2 สามารถนำมาเขียนโปรแกรมได้โค้ดที่สั้น และสามารถนำฟังก์ชันที่เขียนไปใช้ในการพัฒนาโปรแกรมอื่น ๆ ต่อไปได้ ผู้ที่เขียนโปรแกรมคลิปเปอร์จะมีฟังก์ชันที่เขียนเอาไว้มากขึ้นเรื่อย ๆ ทำให้การพัฒนาโปรแกรมต่อ ๆ ไปง่ายขึ้นและเสร็จเร็วขึ้นมาก สำหรับผู้เขียนยังมีฟังก์ชันอื่น ๆ ที่เขียนไว้อีกมาก และจะนำมาเสนอต่อไป ส่วนท่านที่มีฟังก์ชันดี ๆ เขียนมาแบ่งกันใช้บ้างชิครับ